Sborník vědeckých prací Vysoké školy báňské - Technické univerzity Ostrava
číslo 2, rok 2005, ročník LI, řada strojní
článek č. 1484

Tibor SZAPPANOS[*], Iveta ZOLOTOVÁ[*], Lenka LANDRYOVÁ[**]

DISTIRIBUTED DATA MINING AND DATA WAREHOUSE

DISTRIBUOVANÉ DOLOVANIE DÁT A DÁTOVÝ SKLAD

Abstract

This article deals briefly about distributed data mining in data warehouse. It further considers the possibilities of applications under industrial conditions, perhaps in connection with modifications of some Distributed Decision Tree Algorithm.

Abstrakt

Tento článok popisuje distribuované dolovanie dát z dátového skladu. Sú tu diskutované problémy a vývoj distribuovaného dolovania dát v priemeslných podmínkách. Diskutujeme o problémoch učenia sa z distribuovaných dát– algoritmus rozhodovacieho stromu.

1 DATA MINING IN DATA WAREHOUSE

Data mining technology has emerged as means for identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand. Most tools operate on a principle of gathering all data into a central site, then running an algorithm on that data. Distributed data mining refers to the mining of distributed data sets. The data sets are stored in local databases, hosted by local computers, which are connected through a computer network. Data mining takes place at a local level and at a global level where local data mining results are combined to gain global findings [1, 6].

2 PROBLEM OF DATA MINING IN INDUSTRY

By processing data from car purchasing and by subsequent constructing new prices companies can run across decision - making problems on how to correctly apply constructing prices to sell utmost and to have the highest profit. Data from sales and the price structure of particular cars – the prices of different components and various discounts – are stored in diverse archives and because of the easier accessibility, also in a datawarehouse. There are vertically distributed data structures, where the instances are represented by the couple attribute – value. Data in this set can contain errors or attribute values can be missing. For the decision - making for constructing the price of these products there have to be generated clear rules and there have to be found clear indications, which ones are important for product classifications and for sales into classes and the subsequent prediction of new prices. The target function has to have discrete outputs and a classification algorithm shall be applied with the classification without the need of too much computation and it shall be capable of working with continual and categorical variables at the same time.

* Ing., Department of Cybernetics and Artificial Intelligence, FEI Technical University of Košice, Letná 9, Košice, tel. (+421) 55 625 35 74, e-mail tibor.szappanos@skoda-auto.cz
* doc. Ing., CSc., Department of Cybernetics and Artificial Intelligence, FEI Technical University of Košice, Letná 9, Košice, tel. (+421) 55 602 25 51, Iveta.Zolotova@tuke.sk
**Ing., CSc., Department of Control Systems and Instrumentation, FME VŠB - Technical University Ostrava, 17. listopadu 15, Ostrava, lenka.landryova@vsb.cz

A suitable way on how to handle this problem is to use a tree distributed classifier, which is published in detail in [2, 3, 4].

3 DISTRIBUTED DATA STRUCTURE

In a distributed setting, as proposed by [4], the data are distributed across several data sources. Each data source contains only a fragment of the data. This leads to a fragmentation of a data set D. Two common types of data fragmentation are (Figure 1): horizontal fragmentation wherein (possibly overlapping) subsets of data tuples are stored at different sites; and vertical fragmentation (Figure 1), wherein (possibly overlapping) sub-tuples of data tuples are stored at different sites. More generally, the data may be fragmented into a set of relations (as in the case of tables of a relational database, but distributed across multiple sites).

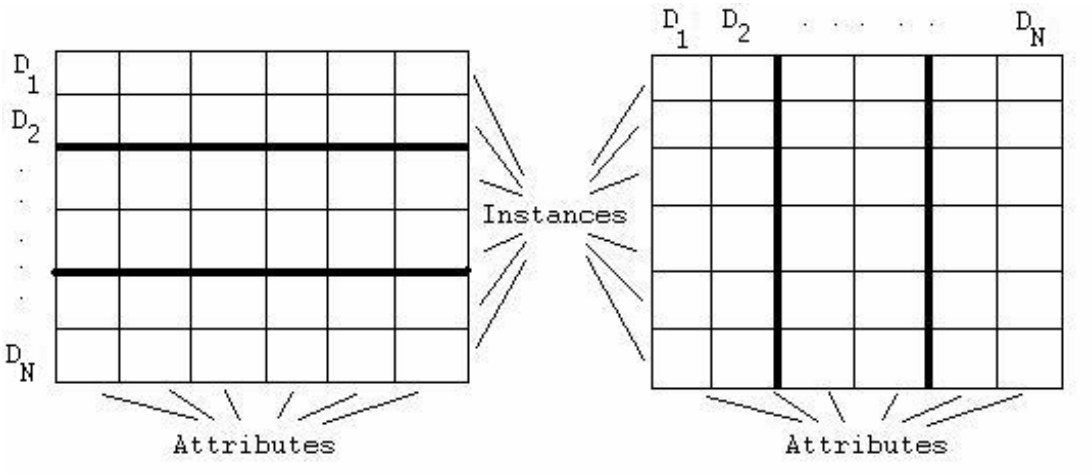

Fig. 1 Horizontally and vertically fragmented data

The problem of learning from distributed data can be summarized as follows: Given the fragments D1, … , DN of a data set D distributed across the sites 1, … , N, a set of constraints Z, a hypothesis class H, and a performance criterion P, the task of the learner Ld is to output a hypothesis $h \in H$ that optimizes P using only operations allowed by Z. As in the case of centralized learning, this is likely to result in a classifier that can be used to classify new unlabeled data. Clearly, the problem of learning from a centralized data set D is a special case of learning from distributed data where K = 1 and Z = φ. This problem is solved in detail e.g. in [1, 2, 3, 4, 6].

4 LEARNING DECISION TREE CLASSIFIERS FROM DATA

Several learning algorithms have been developed, e.g. eager learning algorithms (Naive Bayes Algorithm, Decision Tree Algorithm, Perceptron Algorithm, Support Vector Machines) and also a well known lazy learning algorithm (k Nearest Neighbors).

Learning Phase

ID3(D,A) //D is a set of training examples, A is a set of attributes.//
Create a Root *node for the tree.*
If (*all the examples in* D *are in the same class* c_i)
{
    return (*the single node tree* Root *with label* c_i)
}
else
{
    Let a *be the* Best Attribute (D)
    for (*each possible value* v *of* a) **do**
    {
        Add a new tree branch below Root, *which corresponds to the test* a=v.
        if (D_v *is empty*)
        {
            Below this branch add a new leaf node with label equal to the most common class value in D.
        }
        else
        {
            Below this branch add the subtree ID3(D_v, A-a).
        }
    }
}
return Root.
end learning phase

Classification Phase

Given a new instance **x**, *use the tree having* Root *to clasify* **x**:

- *Start at the root node of the tree, while testing the attribute specified in this node,*
- *Move down the tree branch, which corresponds to the value of the attribute in the given example,*
- *Repeat the procedure for the subtree rooted at the new node till this node is a leaf providing the classification of the instance.*

Fig. 2 Basic preudocode of decision tree [4]

Decision tree algorithms [4, 7] are among some of the most widely used machine learning algorithms for building pattern classifiers from data. The ID3 (Iterative Dichotomizer 3) algorithm [7] and its more recent variants represent a widely used family of decision tree learning algorithms. The ID3 algorithm searches in a greedy fashion, for attributes that yield the maximum amount of information for determining the class membership of instances in a training set D of labeled instances. The result is a decision tree that correctly assigns each instance in D to its respective class. The construction of the decision tree is accomplished by recursively partitioning D into subsets based on values of the chosen attribute until each resulting subset has instances that belong to exactly one of the H classes. The selection of an attribute at each stage of construction of the decision tree maximizes the estimated expected information gained from knowing the value of the attribute in question. The basic pseudocode (learning and classification phase) is published e.g. in [4].

Decision trees are eligible for getting the knowledge and generating decision rules from the data storage based on the following features:

- Decision trees are eligible to generate comprehensive rules,
- Reaches classification without a need of intensive computing,
- Are eligible to work with both continuous and categorical variables, provides with clear indication, which areas are the most important for prediction or classification,

Learning by the use of decision trees is commonly the most eligible for solving problems with characteristics, when instances are represented by two attributes – value, target function has discrete output values, problems requiring disjunctive description, values for training content errors, training data can contain missing attribute values.

4.1 Decisions Tree - Horizontally Fragmented Distributed Data

In [4] is shown the sufficient statistics for learning exact decision trees from horizontally and vertically fragmented data. When the data are horizontally distributed, examples corresponding to a particular value of a particular attribute are scattered across different locations. In order to identify the best split at a node in a partially constructed tree, all the sites are visited and the counts corresponding to candidate splits of that node are accumulated. The learner uses these counts to find the attribute that yields the best split to further partition the set of examples at that node. Most of the pseudocode in Figure 2 remains the same, except that it expands the procedure that determines the best attribute for a split, by showing how to do this when data are horizontally distributed.

From point of view of time complexity - if both serial and parallel access to the data is allowed - parallel access is preferred as it results in an algorithm for learning Decision Tree classifiers from horizontally distributed data that is N faster than the algorithm for learning Decision Tree Classifiers from centralized data.

From point of view of communication complexity, under the assumption that each data source allows shipping of raw data and computation of sufficient statistics, the algorithm for learning decision trees from horizontally distributed data, (where we ship the statistics all the time) is preferable to the algorithm for learning from centralized data.

4.2 Decisions Tree - Vertically Fragmented Distributed Data

When the data are vertically distributed, similar to the algorithm for learning decision trees from horizontally fragmented, the algorithm for learning from vertically distributed data [4] applies a refinement operator R_{DT} to refine a partial decision tree h_i based on the sufficient statistics collected from $D_1, \ldots, D_n$ given h_i. As can be seen, the distributed algorithm is similar to the centralized algorithm, except the part where the best attribute is chosen from the candidate attributes.

From point of view of time complexity - if both serial and parallel access to the data is allowed - then parallel access is preferred as it results in an algorithm for learning Decision Tree classifiers from vertically distributed data, that is N faster than the algorithm for learning Decision Tree classifiers from centralized data, shown in Figure 2.

From point of view of communication complexity, under the assumption that each data source allows, shipping of raw data and computation of sufficient statistics, the algorithm for learning decision trees from vertically distributed data (shown in algorithm, where we ship the statistics all the time) is preferable to the algorithm for learning from centralized data, shown in terms of communication complexity.

5 CONCLUSION

Distributed data mining intends to get the global knowledge from the local data at distributed sites – tables. Data in data storage are distributed into different tables: fact table and dimension table.

Getting the knowledge by the use of decision trees is highly focused because of their ability to work with data, which are not complete or contain errors. Decision trees are eligible for generating unambiguous and comprehensive rules from data stored in the data storage and by doing this to support the decision making – e.g. after constructing a decision tree from particular data storage the company managers are able to find the optimal decisions for managing of a company or its part much easier or they can predict the influence of their decisions.

That's just the reason why the trees and distributed trees are suitable for application of the presented industrial data in the automotive industry in contrary to other methods. In future, it will be of advantage to apply and evaluate decision distributed tree algorithms [4] or extend them for other algorithms available for our industry data (e.g. CART, CHAID – [8]).

REFERENCES

[1] CLIFTON, Ch. Privacy Preserving Distributed Data Mining. In *13th European Conference on Machine Learning* and *6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)*, Place, Helsinki, Finland, 2002, 19-23.

[2] CARAGEA, D., SILVESCU, A., AND HONAVAR, V.: A Framework for Learning from Distributed Data Using Sufficient Statistics and its Application to Learning Decision Trees. International Journal of Hybrid Intelligent Systems. Vol. 1, No. 2., 2004, pp. 80-89.

[3] CARAGEA, D., SILVESCU, A., AND HONAVAR, V.: Decision Tree Induction from Distributed Data Sources. In: *Proceedings of the Conference on Intelligent Systems Design and Applications (ICDA 2003)*, August 10-13, 2003, Tulsa, OK, USA. Pp. 341-350. Springer-Verlag.

[4] CARAGEA, D. (2004) Learning classifiers from distributed, semantically heterogeneous, autonomous data sources. Ph.D. dissertation. Department of Computer Science, Iowa State University, Ames, Iowa, 50011. USA. Available online from URL <http://www.cs.iastate.edu/~dcaragea/thesis.pdf>.

[5] SZAPPANOS, T. Object Data Model and Its Utilization in DSS. In *Proceedings of 2nd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, SAMI 2004*, Place : Herľany, Slovakia, 2004, pp. 111-118, ISBN 963 7154 23 X.

[6] FU, Y. Distributed Data Mining: An Overview. *Newsletter of the IEEE Technical Committee on Distributed Processing*, Spring, 2001, pp. 5-9.

[7] QUINLAN, R.: Introduction of decision trees. *Machine Learning,* 1:81–106, 1986.

[8] Wilkinson, L.: Tree Structured Data Analysis: AID, CHAID and CART. Paper presented at the 1992 Sun Valley, ID, Sawtooth/SYSTAT Joint Software Conference. Available online from URL <http://www.spss.com/research/wilkinson/Publications/c&rtrees.pdf>

Reviewer: doc. Ing. Radim Farana, CSc., VŠB-Technical University of Ostrava